CLAIMS:

Page 14, line 2, replace [White Space Algorithm] with method for locating white space

The following listing of all claims in the application with their status and the text of all active claims.

1. (Currently Amended) A method for supporting the automatic creation of ~~high quality~~ computer-based drawings that are virtually free from graphics and text overwrites, comprising:

   a series of drawing plane bitmaps constructed by stroking graphics and text elements into an equivalent pixel ~~sets~~ set that represents committed drawing graphics and a separate equivalent pixel set that represents proposed drawing graphics;

   the use of said bitmaps to dynamically track the ongoing state of committed drawing graphics;

   the use of said bitmaps to compare proposed drawing graphics with already committed drawing graphics to ~~determine the quality of white space available~~ confirm the availability of white space for the placement of proposed new graphics or to determine the degree of overwriting;

   and based on the availability of white space or the degree of overwriting, ~~the use of~~ control variables and status variables are used to guide adjustments along a vector, or about a rotation point, as a means of moving proposed new graphics into ~~locations of white space.~~ either a location of white space or into a location with less overwriting.

2. (Currently Amended) The method according to claim 1, wherein bit level manipulations are used to track and compare committed drawing and proposed drawing bitmaps to quantify the quality of white space available in the vicinity of tentative placement coordinates.

3. (Currently Amended) The method according to claim 1, wherein the said control variables enable client applications to activate and manipulate the behavior of the ~~white space algorithm~~ adjustments.

4. (Currently Amended) The method according to claim 1, wherein the said status variables quantify the result of tracking and comparison operations and provide client applications with essential information to determine the quality of white space which is quantified by the degree of overwriting.

5. (Currently Amended) The method according to claim 1, wherein primary, secondary and supplementary slide vectors define the direction along which adjustments are made to move a proposed tentative graphics element into a location of white space or location with less overwriting.

6. (Currently Amended) The method according to claim 1, wherein a point at a given radius to a given coordinate and a rotation about which adjustments are made to move a proposed tentative graphics element into a location of white space or location with less overwriting.

7. (Previously Presented) The method according to claim 3, wherein said control variables include halos around graphics elements for controlling the proximity of proposed tentative graphics to committed graphics.

8. (Currently Amended) The method according to claim 1, wherein the method ~~white space algorithm~~ operates as a server type nucleus that ~~can be utilized by a plethora of different client applications from virtually any discipline.~~ is utilized by client applications that are applied to solve a class of problems where engineering drawings, architectural drawings, maps or other types of diagrams are composed automatically by computer software applications such as, but not limited to, engineering design algorithms, architectural design algorithms and a database of spatial relationships.

9. (Previously Presented) The method according to claim 3, wherein said control variables allow a skip over feature to be enabled in order to suppress the drawing of a portion of a line over dense areas of committed graphics.

10. (Currently Amended) The method according to claim 4, wherein said status variables ensure the complete generality of the algorithm and provide information to client applications that ~~can be~~ are applied to solve a class of problems where engineering drawings, architectural drawings, maps or other types of diagrams are composed automatically by computer software applications such as, but not limited to, engineering design algorithms, architectural design algorithms and a database of spatial relationships.

11. (Currently Amended) The method according to claim 2, wherein areas of available white space, adequate to position new graphical objects, can be efficiently identified with low level computer instructions, comprising:

    the use of LOGICAL OR and LOGICAL AND operations that make efficient use of computer resources;

    and the tracking of areas occupied by previously positioned graphical objects with low level logical OR instructions.

    ~~(Logical AND). In addition, areas occupied by previously positioned graphical objects are efficiently tracked with low-level computer instructions (logical OR). By employing such logical operations this method makes efficient use of computer resources, as is well known to those skilled in the art.~~

12. (Currently Amended) The method according to claim 1, wherein text elements are represented in the bitmap as ~~a set of parallel lines and are~~ an equivalent filled rectangular block and is used for the purpose of overwrite detection.

13. (Currently Amended) The method according to claim 1, wherein the method provides performance levels that are linear and proportional to the total number of graphics elements in the completed drawing. ~~Therefore the method is scalable.~~

14. (Currently Amended) The method according to claim 1, wherein white space is located through the use of efficient bitmap manipulations to compare the additional graphics with already committed graphics ~~to provides~~ in a manner that ensures performance levels for comparisons ~~that~~ are independent of the complexity and density of the committed graphics.

15. (Previously Presented) The method according to claim 1, wherein the said bitmaps are also used to represent a 'must hit' filled polygon region that is used to ensure maximum overlap for the proposed graphics, achieved by inverting the objective of the bit level comparisons in favor of an overlay.

16. (Previously Presented) The method according to claim 12, wherein the said text elements that contain more than a single word, may be split into equivalent multi-line text elements when sufficient white space cannot be located for the initial single line text element.